ALMA MATER STUDIORUM
UNIVERSITÀ DI BOLOGNA
CAMPUS DI FORLÌ

91258 / B0385
Natural Language Processing

Lesson 17. Recurrent Neural Networks

Alberto Barrón-Cedeño
a.barron@unibo.it

24/11/2025

Previously

- CNNs for text

Table of Contents

Chapter 8 of Lane et al. (2019)

Introduction

Introduction

CNNs

- Adequate for processing *full* texts (∼sentences)
- Words tending to appear close to each other are spotted and play a joint role
- Longer relationships —farther than $[3, 4]$ words are ignored

What is missing?

- Keeping track of what happened long ago
- Memory

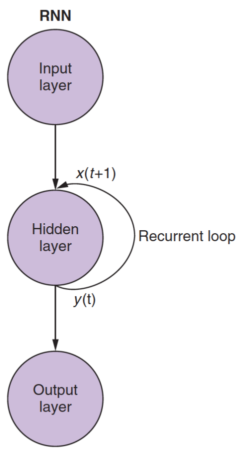
- Language is not an image —no snapshots
- Language is a sequence; both text and speech

Keeping the past *in mind*

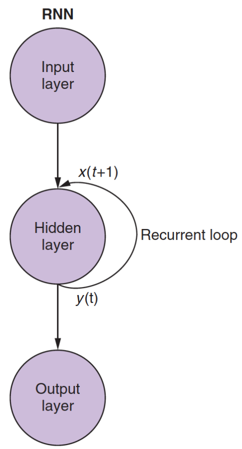
Remembering the Past

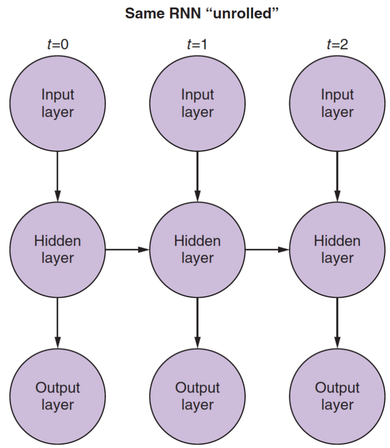
$$w_0\ w_1\ w_2\ w_3\ \dots\ w_{t-1}\ w_t\ w_{t+1}$$

- To understand a text at time t, we need to consider what happened at time $t-1$, $t-2$, ..., $t-k$
- Recurrent neural nets (RRN) come into play
- RNNs combine what is happening now with what happened earlier



Full feed-forward networks that consider their own output

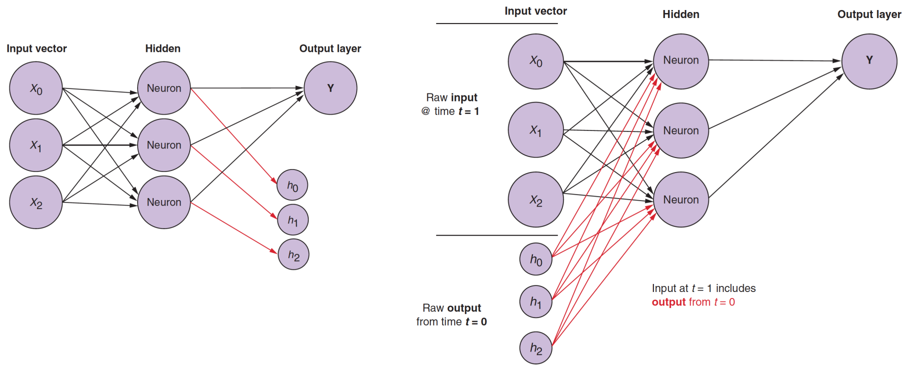




* All three columns represent the same network (at different times)

(Lane et al., 2019, p. 252)

Zooming into the *unrolled* RNN: t and $t+1$

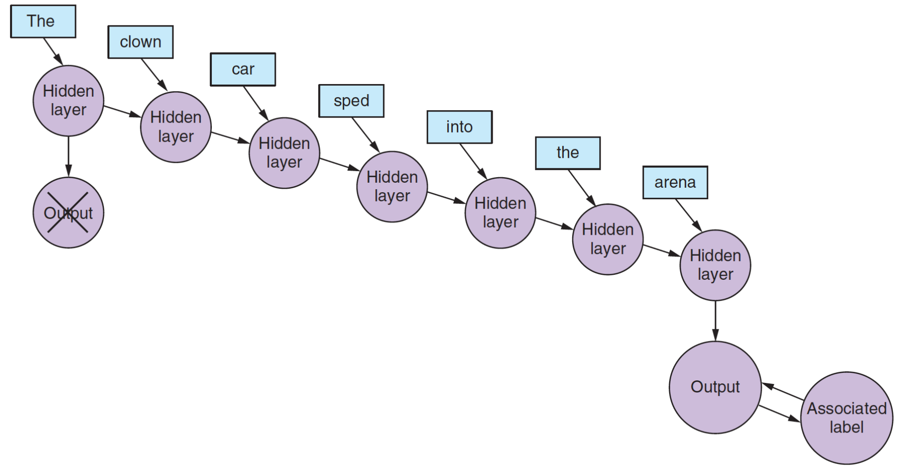


$t = 0$ $\quad$ $t = 1$

- The red arrows are just *standard* connections, with weights
- Now we can feed the text, one word at a time

(Lane et al., 2019, p. 252–253)

"Multiple inputs, one output"

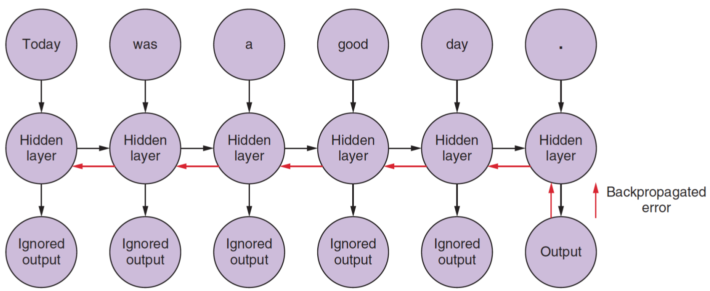


- A network for variable input lengths (lengthy inputs be troublesome)
- Rather than *mutually-independent* snapshots,[1] there is a sense of time

[1] With overlaps, not as bad as it sounds

(Lane et al., 2019, p. 254)

Backpropagation through Time: the "Vanilla" Way

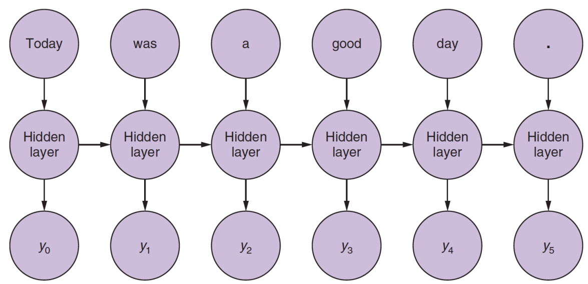



```
error = y_true_label - y_output
```

- All intermediate outputs are ignored; the loss is computed at the end
- The same chain rule is applied to do backpropagation; but this time it heads to "the past"
- The weight corrections are calculated for each t
- The combined updates are applied only until reaching $t = 0$

(Lane et al., 2019, p. 256)

Backpropagation through Time: the Better Way




```
error = sum([y_true_label[i] - y[i] for i in range(6)])
```

- We compute the loss combining all intermediate outputs
- The weight corrections are still additive: the update is applied until
  1. computing all errors and
  2. reaching back to the weight adjustments in $t = 0$

Let us see

(Lane et al., 2019, p. 258)

RNNs in Keras

RNN in Keras: what we have so far

We have setup a simple recurrent neural network

- The input sequences have fixed length: 400 tokens (each 300D)
- Our recurrent layer contains 50 units
- The output will be 400×50:
  - 400 elements
  - one 50D vector each

`return_sequences=True`

True return the network value at each t: 400 50D vectors

False return a single 50D vector (default)

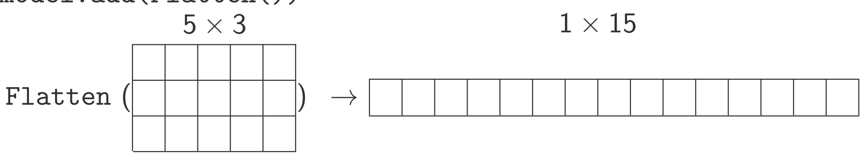
True → this is why we are padding

Let us see

RNN in Keras: further details

- A `Dense` layer expects a *flat* vector

```
model.add(Flatten())
```



- In our case: $400 \times 50 \rightarrow 1 \times 20,000$

Let us see

Example derived from
https://stackoverflow.com/questions/43237124/role-of-flatten-in-keras

Some parameters are "free"

embedding_dims comes from the embedding space; hard to change, but possible: other embeddings, 1-hot

num_neurons kind of arbitrary; can be changed

maxlen kind of arbitrary; can be changed (or neglected)

batch_size bigger→faster (higher local minimum risk)

epochs trivial to increase (avoid starting from scratch each time)

Let us see

Important: unless you have access to HPC, don't go *bananas* when exploring parameters (and perhaps even in that case)

Try some sensitive configurations and keep track of all the settings and outputs[2]

[2] See, for instance, Fernicola et al. (2020).

References

Fernicola, F., S. Zhang, F. Garcea, P. Bonora, and A. Barrón-Cedeño
2020. Ariemozione: Identifying emotions in opera verses. In *Italian Conference on Computational Linguistics*.

Lane, H., C. Howard, and H. Hapkem
2019. *Natural Language Processing in Action*. Shelter Island, NY: Manning Publication Co.